

## Nivel Básico

### 1) Suma total

Dado un array de números, obtené la suma de todos sus elementos.

Entrada de ejemplo: [2, 3, 5]

Salida esperada: 10

Pista: Acumulá un número que empieza en 0.

### 2) Producto total

Calculá el producto de todos los números de un array.

Entrada de ejemplo: [2, 3, 4]

Salida esperada: 24

Pista: Empezá en 1 y andá multiplicando.

### 3) Concatenación con separador "-"

Uní todas las palabras de un array en un solo string separados por "-".

Entrada de ejemplo: ["hola", "mundo", "js"]

Salida esperada: "hola-mundo-js"

Pista: Cuidá no agregar "-" adelante del primero.

### 4) Cantidad de ocurrencias

Contá cuántas veces aparece cada fruta y devolvé un objeto { fruta: cantidad }.

Entrada de ejemplo: ["pera", "manzana", "pera", "uva"]

Salida esperada: { pera: 2, manzana: 1, uva: 1 }

Pista: Usá un objeto y sumá 1 por clave.

### 5) Eliminar duplicados

A partir de un array con repetidos, devolvé un array sin duplicados (mismo orden de primera aparición).

Entrada de ejemplo: [3,3,1,2,1]

Salida esperada: [3,1,2]

Pista: Si ya lo viste, no lo agregues.

## Nivel Intermedio

### 6) Aplanar un nivel

Aplaná un array de arrays (una sola profundidad) en un único array.

Entrada de ejemplo: [[1,2],[3],[4,5]]

Salida esperada: [1,2,3,4,5]

Pista: Concatená cada sub-array a un acumulador [].

### 7) Índice por id

Convertí un array de objetos en un índice por id: { [id]: objetoCompleto }.

Entrada de ejemplo: [{id:1,n:"A"},{id:2,n:"B"}]

Salida esperada: { 1:{id:1,n:"A"}, 2:{id:2,n:"B"} }

Pista: Acumulador {}, asigná acc[obj.id] = obj.

### 8) Máximo por propiedad

Dada una lista de empleados, obtené el empleado con mayor salario.

Entrada de ejemplo: [{n:"Ana",s:50},{n:"Luis",s:80},{n:"Sol",s:70}]

Salida esperada: {n:"Luis",s:80}

Pista: Guardá el "mejor hasta ahora".

### 9) Particionar pares e impares

Separá números en un objeto con dos arrays: { pares:[], impares:[] }.

Entrada de ejemplo: [1,2,3,4,5]

Salida esperada: { pares:[2,4], impares:[1,3,5] }

Pista: Elegí el array correcto y push.

### 10) Histograma de letras

Dado un string, devolvé un objeto con la cantidad de veces que aparece cada letra (ignorando espacios y en minúsculas).

Entrada de ejemplo: "Hola Hola"

Salida esperada: { h:2, o:2, l:2, a:2 }

Pista: Pasá a minúsculas, quitá espacios y contá.

## Nivel Avanzado

### 11) Totales por categoría

Dado un carrito con {categoria, precio}, calculá el total gastado por categoría.

Entrada de ejemplo: [{c:"libros",p:100},{c:"tech",p:200},{c:"libros",p:50}]

Salida esperada: { libros:150, tech:200 }

Pista: Sumá precio por categoria.

### 12) Promedios por grupo (una pasada)

Dada una lista de notas {curso, nota}, devolvé {curso: promedio} sin recorrer dos veces (acumulá sumas y conteos en el mismo reduce).

Entrada de ejemplo: [{c:"A",n:8},{c:"B",n:6},{c:"A",n:10}]

Salida esperada: { A:9, B:6 }

Pista: Acumulá {suma, cuenta} por curso y al final calculás suma/cuenta.

### 13) Pipeline de funciones

Dada una lista de funciones unarias y un valor inicial, aplicá todas en cadena con reduce (composición izquierda→derecha).

Entrada de ejemplo: `fns=[x=>x+1, x=>x*2]`, `valor=3`

Salida esperada: 8

Pista: El acumulador es el valor actual transformado.

### 14) Verificador de paréntesis

Con reduce, verificá si una cadena de paréntesis está balanceada. Si en algún momento el conteo baja de 0 o termina distinto de 0, es inválida.

Entrada de ejemplo: `"(())"` y `"()(""`

Salida esperada: true y false, respectivamente

Pista: `"("` suma 1, `")"` resta 1.

### 15) Suma acumulada (prefix sum)

Generá un nuevo array con la suma acumulada de los elementos.

Entrada de ejemplo: `[3,1,4,2]`

Salida esperada: `[3,4,8,10]`

Pista: El siguiente valor es previo + actual.